Тестовый вариант программы приемника разработан для обработки данных с одного датчика. Но при серийном и массовом производстве прибора необходимо изменение программы приемного модуля, при котором он сможет принимать и обрабатывать данные с четырех колес в мультиплексном режиме.

Технические характеристики

- Порог срабатывания сирены, кПа……..……190
- Максимальное количество датчиков, подключаемое к системе…..4
- Разрядность преобразования АЦП, бит....……..8
- Скорость обмена данными, бит/с……200
- Питание передающей части – батарея CR2032, напряжение 3 В
- Питание приемной части – от бортовой сети автомобиля 12В

Порядок работы

Передающая часть включена постоянно, приемная часть включается вместе с появлением напряжения в бортовой сети автомобиля, т.е. при включенном двигателе. Нажатием на кнопку на лицевой панели включается режим мониторинга давления. Об этом свидетельствует свечение светодиода «Мониторинг».

Водитель начинает движение на автомобиле. При опасном снижении давления ниже 1,9 атмосферы в любом из четырех колес приемник начинает издавать прерывистый громкий звук из встроенного динамика и мигать светодиодом «Авария», что привлекает внимание водителя и он останавливает автомобиль, согласно правилам дорожного движения. Мигание и звук прекращаются при отключении режима мониторинга т.е. при нажатии кнопки. Далее водитель восстанавливает колесо, либо меняет его на запасное, в этом случае необходимо переставить датчик давления на новое колесо. Затем садится в автомобиль, нажимает кнопку на приемнике и продолжает движение.

Литература:

1. Яценков В.С. Микроконтроллеры Microchip® rfPIC™ со встроенным маломощным радиопередатчиком, издание первое, 2006г., Горячая линия – Телеком, 344 стр.
2. http://www.microchip.ru/

РЕАЛИЗАЦИЯ СИСТЕМЫ РАНЖИРОВАНИЯ ДОСТУПА К ФУНКЦИЯМ УСТРОЙСТВА С ПОМОЩЬЮ АППАРАТНЫХ КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ.

Автор: Молоталиев Александр Олегович, студент 4-го курса.
Руководитель: Коковин Валерий Аркадьевич, кандидат технических наук, доцент кафедры «Автоматизация технологических процессов и производств».
Образовательное учреждение: филиал «Протвино» Международного университета природы, общества и человека «Дубна», г. Протвино.

Устройство и особенности работы криптографических ключей

Криптографический аппаратный ключ представляет собой устройство способное помимо хранения данных производить манипуляции над ними[5]. Первое поколение устройств данного типа представляло из себя микроконтроллер общего назначение с расширенными возможностями коммуникации с персональным компьютером. Особенностью ключей данного поколения является низкая устойчивость системы защиты на их основе и возможность создания дампа памяти при имеющихся средствах разработки ПО микроконтроллеров и аппаратных отладчиков.

Второе поколение отличается от первого наличием протокола обмена между компьютером и ключом. Также они требуют наличия драйверов в системе для взаимодействия с клиентским ПО. Третье поколение, помимо наличия протокола обмена имеет еще набор прикладных интерфейсов, необходимых для работы с компонентами ключа.

Четвертое и пятое поколение имеет возможность удалённого обновления данных в ключе.

За время существования, ключи взаимодействовали с ПО через разные внешние интерфейсы ввода –вывода. Наиболее распространённым в настоящее время, в виду его распространённости и удобства сопряжения устройства, является USB, но раньше большинство ключей работало через LPT порт.

Помимо вышеперечисленных видов и интерфейсов подключения на данный момент существуют так называемые программные ключи защиты. По своему устройству они представляют обычный флэш

накопитель, но имеющий специальное ПО. Основное назначение данной разновидности системы защита мобильных приложений для планшетных компьютеров на базе Windows 8 SL/PRO или Android.

На данный момент распространены следующе бренды ключей.

2. Guardant
3. SenseLock
4. HASP

Размер памяти ключа, доступной на чтение-запись, обычно составляет не более 126 байт. Часть этой памяти используется для унификации ключа. В ней хранится некоторая служебная информация, тип ключа, идентификатор пользователя (Customer's code), Серийный номер ключа (ID-number). Код пользователя доступен только на чтение. Это достигается с помощью специальной процедуры программирования через дополнительный адаптер. Память в ключе организуется полями и требует дополнительной процедуры предварительного форматирования. Причем, запись в память ключа будет невозможна, если заранее не были правильно определены типы полей, их размеры и адреса.

На текущий момент самый привлекательный тип электронных ключей – ASIC[5], т.к. все особенности их функционирования закладываются на стадии проектирования так называемого ASIC-чипа (заказной интегральной микросхемы специального применения). Как правило такой чип имеет достаточно сложную внутреннею организацию и нетривиальные алгоритмы работы. Логику работы такого чипа практически невозможно реализовать с помощью стандартных наборов микросхем PAL (Programmable Array Logic), GAL. Такой чип практически невозможно воспроизвести, а содержащийся в его памяти микрокод - считать и расшифровать либо эмулировать. По сути ASIC чип представляет собой абстрактный цифровой автомат с памятью.

Перед изготовлением ключа под конкретного заказчика сначала с помощью специальной аппаратуры программируется ASIC-чип. В чип заносится уникальный код, присвоенный этому заказчику, и, может быть уникальный серийный номер изготавливаемого ключа. Эта информация будет доступна в дальнейшем ТОЛЬКО на чтение и ее нельзя будет изменить никакими средствами. Другим, менее надежным способом, является программирование готового ключа с "чистой" микросхемой. Такое программирование обычно осуществляется с помощью специального адаптера или вставляемой в слот компьютера платы-программатора. Некоторые типы ASIC-чипов имеют блокировку доступа (ключ для программирования).

Одним из главных критериев качества защиты является устойчивость к эмуляции. Эмуляторы электронного ключа могут быть как программные, так и аппаратные. При этом эмулироваться может как протокол обмена ключа с портом, так и сам ключ. Ключи, построенных на базе EEPROM-памяти[5], используют для защиты от эмуляции так называемые "плавающие" или изменяемые и "зашумленные" протоколы обмена с портом. Ключи на базе ASIC-чипа имеют дополнительную защиту от эмуляции, реализованную в виде сложной функции, статистический анализ которой невозможно провести за приемлемое время.

ASIC-чип обычно реализует некоторую сложную функцию y=f(x), где x - данные, передаваемые ключу из программы, y - данные, возвращаемые ключом в программу, f(x) - функция преобразования входных данных в выходные.

Одной из разновидностей реализации функции f(x) является использование ASIC-чипа в качестве внешнего вычислителя (по отношению к процессору компьютера. К этому методу прибегают для получения из ключа некоторой значащей информации, влияющей на ход выполнения программы и для избавления проверок на защиту. Например, несколько функций жизненно важных для функционирования программы, реализуются аппаратно с помощью ключа и отсутствуют в теле программы. Из ключа могут также возвращаться смещения по которым выполняемой программой передается управление или считываются данные.

Электронные ключи, выполненные на базе микропроцессора[5], являются платформо-независимыми. Внутренний микропроцессор ключа реализует некий сложный алгоритм преобразования данных. При работе защищенное приложение посылает ключу стандартный запрос. Ключ обрабатывает этот запрос и по заданному алгоритму выполняет некие преобразования данных.

Микропроцессорные ключи обычно поставляются разработчикам программ "чистыми" с исходными кодами процедур доступа к ключу, так что разработчики сами могут их программировать в той среде или на той платформе, для которой пишется их приложение. Протокол обмена между ключом и компьютером (рабочей станцией) динамически шифруется. Доступ к ключу защищается специальным паролем, назначаемым самим разработчиком программного обеспечения.

Для хранения информации о номере версии программы, ключей доступа, шифрования, счетчиков память EEPROM. Кроме того, многие модели ключей имеют RAM-память для временного хранения переменных, результатов некоторых вычислений и т.д.

Примеры приложений использующие аппаратные ключи:

5. 1С Бухгалтерия
6. ORACLE Sybase

Разработка алгоритма взаимодействия клиентской программы и ключа

В данной работе приводится разработанный алгоритм взаимодействия программы и аппаратного ключа.

- Клиентская программа при загрузке ищет ключи, если ключ найден то программа продолжает работу иначе завершает выполнение.
- Из памяти ключа программа на ПК читает данные пользователя и маски прав доступа
- При первом использовании записывает данные пользователя в микроконтроллер устройства, в случае ошибки выводиться соответствующее сообщение
- Перед записью параметров происходит проверка наличия ключа и вычисления контрольной суммы по алгоритму CRC[5].
- Если во время работы ключ был извлечен, то часть функций отвечающих за работы деактивируются
- При каждом подключение ключа происходит проверка прав доступа к компонентам комплекса

Описание клиентского ПО

В качестве примера взаимодействия клиентской программы и ключа приведены графические интерфейсы настройки медицинского прибора - маммографа. На рис.1 показано взаимодействие программыс ключом, при этом ключ правильно определяется и программе разрешается настраивать прибор («Доступ разрешен»).

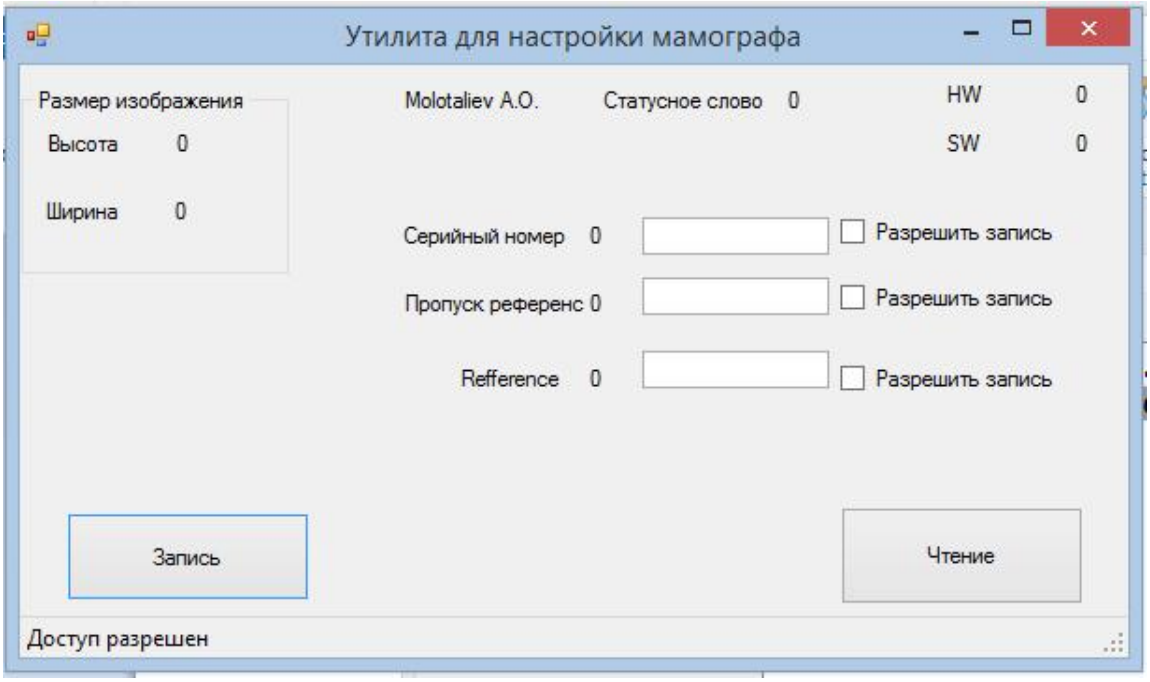

Рис 1.Вид приложения при наличии ключа

На рис.2 программа настройки не обнаружила ключ и поле «запись» не доступно для программы («Доступ ограничен»).

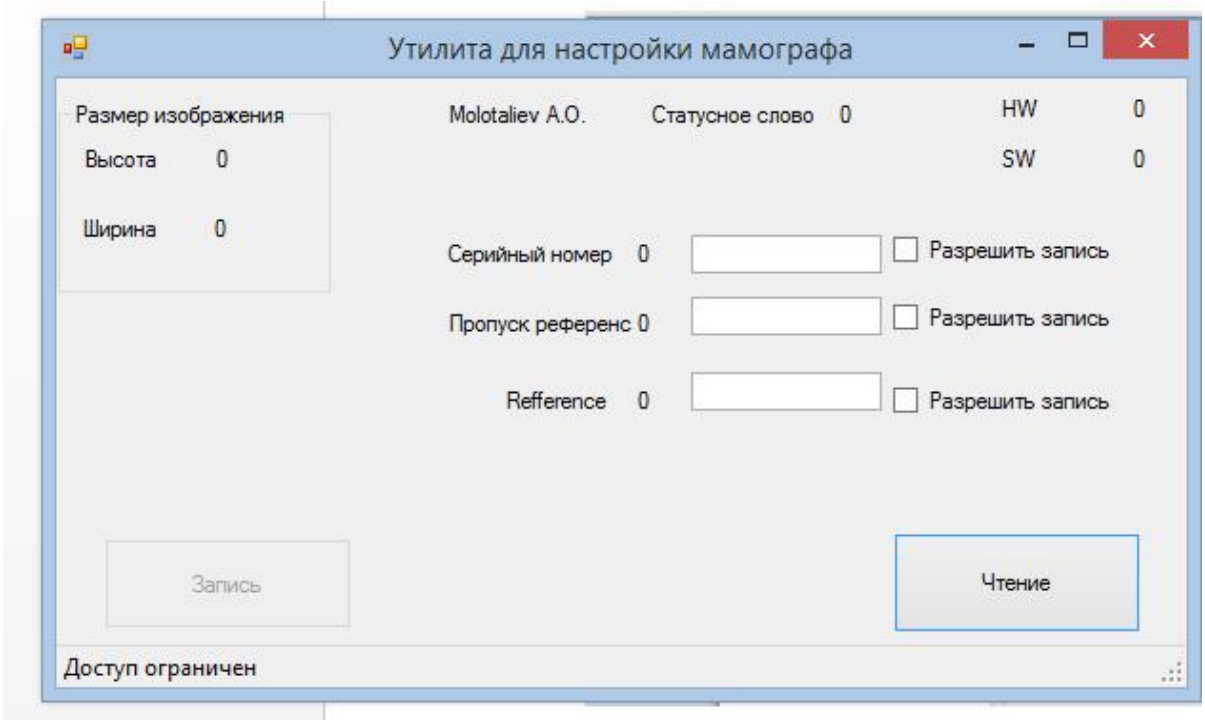

Рис 2 .Вид приложения при отсутствии ключа

Программное обеспечение для ПК написано на C++/CLI[3] с использованием компонентов NET Framework, графический интерфейс основан на технологии Windows Forms[6]. Взаимодействие с ключом происходит через управляемую сборку, а обмен данными происходит с использование классов инкапсулирующих функций интерфейса FTDI.

Для коммуникации с микроконтроллером устройства компьютер пользователя должен обладать FTDI USB – RS485[2] переходником.

Приложение ориентировано на работу в операционной системе Windows, но может быть портировано на Mono для использования в Linux подобных системах.

Чтение данных происходит с помощью API функций ключа[4]. Стоит отметить, что программа не изменяет предварительно запрограммированные значения ячеек в ключе, а только использует их для ограничения прав доступа на запись в память микроконтроллера.

Формат и описание данных ячеек хранящихся ключе

Ключ клиента содержит следующие поля

- Фамилия владельца ключа.
- Количество битовых масок устройств.
- Битовые маски устройств.

В таблице указано распределение памяти по полям

Таблица. Распределение памяти под поля, необходимые для работы системы

Фамилия владельца	16 байт
Количество масок	8 байт
Маска	4 байта

Данные в ключе сохраняются в двоичном представление, а при чтении приводятся в требуемый тип, необходимый для работы программы. В случае отсутствия одного из полей в ключе корректная работа программы не гарантируется. Запись производится с помощью специальной утилиты из комплекта разработчика.

Литература:

1.http://www.guardant.ru - Информация о аппаратном обеспечение ключей используемых в данной разработке

2. http://www.ftdi.com - Информация о коммуникационном модуле

3. Gordon Hogenson – Foundations of C++/CLI – The Visual C++ Language for NET_3.5 / издательство APRESS,2008,USA,497p.

4.Guardant руководство пользователя устройство электронных ключей.

5. Дмитрий Скляров – Искусство защиты и взлома информации/издательство СПб.: БХВ-Петербург,2004. – 288 стр.

6.Мэтью А.Стекер,Стивен ,Джон Стейн,Стивен Нортроп Тони – Разработка клиентских Windows приложений на платформе Microsoft .NET Framework:учебный курс Microsoft / Пер.с англ – М.:Издательство “Русская Редакция”;Спб.:Питер ,2008. – 624 стр.

АЛГОРИТМ ОПРЕДЕЛЕНИЯ НЕОБХОДИМОГО КОЛИЧЕСТВА ЗИП С УЧЕТОМ ВНУТРЕННЕЙ ИЗБЫТОЧНОСТИ АСУ ТП.

Автор: Мурадова Анна Руслановна, студентка 2 курса филиала Московского государственного машиностроительного университета (МАМИ) в г.Серпухове

Научный руководитель: к.т.н., профессор Меша Константин Иванович, заведующий кафедрой «Информационные технологии и управление»

Образовательное учреждение: филиал Московского государственного машиностроительного университета (МАМИ) в г.Серпухове

The algorithm for the determining of a necessary number of SPTE taking into account the inside excess Automatic Process Control System.

Математическая постановка задачи построения ресурсосберегающей стратегии сводится к минимизации целевой функции приращения коэффициента оперативной готовности $\Delta K_{oг}\left(P_{изу}\right)$ в течение времени поддержания АСУ ТП в работоспособности к применению $T_г\left(P_{изу}\right)$, используя внутреннюю избыточность АСУ ТП для восстановления работоспособного состояния, и направлена на разрешение объективного противоречия между старыми неэффективными методами обоснования эксплуатационных процессов и новыми условиями эксплуатации АСУ ТП за пределами гарантийного срока при выделяемых ограниченных ресурсах.

В соответствии с поставленной задачей необходимо оценить экономическую эффективность предлагаемой технологии устранения неисправностей по целевой функции

$$\Delta K_{oг}\left(P_{изу}\right)=\frac{\sum_{i=1}^{n}T_{Oi}}{\sum_{i=1}^{n}T_{Oi}-\sum_{i=1}^{n}T_{Bi}}\cdot P_{изу}\text{, (1)}$$

где T_{Oi} - среднее время наработки на отказ i - го блока;

T_{Bi} - среднее время восстановления i - го блока и ограничении

$$T_г\left(P_{изу}\right)=T_2\left(t_i\right)\cdot\sum_{i=1}^{5}\left[\sum_{i=1}^{n}\left(K_{ai}-k_{ai}\right)+\sum_{i=1}^{m}\left(K_{Фi}-k_{Фi}\right)\right]\text{, (2)}$$

где $T_2\left(t_i\right)$ - среднее время наработки на отказ системы;

K_{ai} - общее количество аппаратно избыточных блоков i - го типа;

$K_{Фi}$ - общее количество функционально избыточных блоков i - го типа;

k_{ai} - минимально необходимое количество аппаратно избыточных блоков i - го типа;

$k_{Фi}$ - минимально необходимое количество функционально избыточных блоков i - го типа.

Другими словами необходимо рассчитать максимальное время работы АСУ ТП с заданной вероятностью безотказной работы, используя в качестве ЗИП при восстановлении готовности заложенную в структуру АСУ ТП аппаратную и функциональную избыточность [1]. Для этого необходимо выполнить следующую последовательность действий.